REMARKS

INTRODUCTION

Claims 1-27 were previously pending and under consideration.

Claims 3, 7, 12, 17, 19, 23, 26, and 27 are cancelled herein.

Therefore, claims 1, 2, 4-6, 8-11, 13-16, 18, 20-22, 24, and 25 are now pending and under consideration.

Claims 1, 2, 4-6, 8-11, 13-16, 18, 20-22, 24, and 25 stand rejected.

Claims 1, 2, 4-6, 8-11, 13-16, 18, 20-22, 24, and 25 are amended herein.

No new matter has been added. Reconsideration and withdrawal of the rejections is respectfully requested.

REJECTION UNDER 35 USC § 103

Claims 1-27 stand rejected under 35 USC § 103 as obvious over Feldmeier in view of Chase. For reasons presented below, reconsideration and withdrawal of the rejection is respectfully requested.

Review of Present Claim Amendments

According to the amended independent claims (1, 6, 11, 16, 20, and 24), an upper layer segment (or PDU) is split, and a split segment (or split PDU) thereof is provided with a remote direct memory location. The split segment's (or split PDU's) remote direct memory location is obtained or determined from the direct memory location of the upper layer segment (or PDU) from which it was split, which allows the split segment (or PDU) to be directly placed in remote memory. Furthermore, the split segment's (or PDU's) remote direct memory location is not placed in the header of the transport segment that carries the split segment, but rather is placed in the body of its transport segment.

Claims 3, 7, 19, 23, 26, and 27 are cancelled because closing a connection when upper layer segments are too large conflicts with splitting the upper layer segments when they are too large, and segment splitting is not usually considered to be an error condition.

Claims 12 and 17 are cancelled as being superseded by the independent claims, which now more broadly recite similar features.

Summary of Rejection

The rejection is based on a combination of Feldmeier and Chase. The general idea of the rejection appears to be that Feldmeier teaches dividing protocol packets into chunks to be carried by another protocol, Chase teaches transporting RDMA communications over TCP, and therefore Feldmeier-Chase teaches dividing RDMA communications into chunks for transport over TCP. As discussed in detail below with relation to the claims, the rejection is traversed because neither Feldmeier nor Chase teach providing RDMA chunks (or split RDMA segments) with remote direct memory locations. This is based on an the fact that (1) the chunks in Feldmeier only have information for placing chunk payloads relative to each other, and (2) Chase fails to teach or suggest any form of chunking, rather, the only relevant teaching of Chase is the renegotiation of MTU size between two virtual interfaces.

Feldmeier: Chunks are relative to each other and do not directly address memory

Feldmeier discusses chunking so that self describing packets and can be processed as they arrive ("Chunks are self-describing units designed for high speed protocol processing ... The self-describing nature of chunks allow them to be processed as they arrive at the receiver regardless of any misordering", Abstract). The "processing" in Feldmeier is not direct memory access, rather the processing in Feldmeier is stack processing. More specifically, "a chunk ... is transmitted with a completely self-describing header. This header contains enough information so that the associated

group of data can be processed by the protocol stack" (column 3, lines 34-41, emphasis added). The "processing" in Feldmeier is nothing more than reassembly of chunks into their original parent chunk. Therefore, the chunk headers in Feldmeier only have relative placement information; information that places the payload of a chunk within the larger unit from which the chunk was split. As discussed below, this is shown by the chunking and reassembly algorithms, the discussion of chunking, and the claims.

Feldmeier's chunking algorithm is included in Appendix A. In the chunking algorithm, original chunk has a length (chunk.len). The original chunk is split or fragmented into new chunks; chunk_a and chunk_b, where the original chunk has a length chunk.len, and the new chunks have smaller length. The new chunks are assigned the same identifiers as the parent chunks, however, the lengths are different. The length of chunk_a is simply the new length; "new_len". The length of chunk_b is the length of the original chunk less new_len (i.e., the remainder of the original chunk). The serial number of chunk_b is incremented (by new_len). The payload of chunk_a is the first new_len bytes of the original chunk, and the payload (data) of chunk_b is the rest of the original chunk's payload. As can be seen, the chunking algorithm in effect merely splits the original chunk into smaller chunks, and the smaller chunks can be ordered by their new serial numbers.

See also Figure 7, where the fragments ("TWO CHUNKS") have the same IDs but different serial numbers. The header of the second new chunk, for example, is clearly an ordinal that is clearly relative to the original chunk.

Appendix B confirms that the fragment or reduced chunks are designed not to be directly placed in memory, but rather are designed to allow a chunk to be immediately used *for reconstructing the parent chunk*, rather than direct memory placement ("we have two chunks, chunk_a and chunk_b, that we wish to reassemble into chunk_c", column 11, lines 25-28).

The claims further confirm this understanding of Feldmeier. Each independent claim clearly states that "a chunk header ... indicates ... the relative position of the basic

data units in said protocol data units" (claim 1, see also claims 11, 15, and 16, which also state that the chunk header indicates a relative position in the protocol data units).

Therefore, it is clear that Feldmeier teaches fragmenting protocol data units into chunks, and the header of each chunk indicates where the chunk's data is relative to the protocol data unit, thus allowing the protocol data unit to be reconstructed without having to wait for all of the chunks and without concern for the order in which chunks arrive.

Applicant respectfully submits that the chunks in Feldmeier are distinctly different than the split segments of the current claims because the split segments of the current claims have information for direct placement into memory. In other words, Feldmeier's chunks have information for placement relative to a parent PDU, whereas the presently claimed split segments have information for placement into a remote memory.

Chase: Does Not Suggest Dividing RDMA PDUs

In section 4.1 ("How RDMA works"), Chase describes the general idea of how "[t]he receiving NIC translates these commands into local memory reads and writes". In other words, the NIC writes directly to host memory rather than passing information to the application layer. Section 4.1 discusses the very basic idea of "embed[ding] new RDMA control commands into the byte stream or packet stream". Chases discusses how to accomplish this in section 6, "Implementing RDMA".

In section 6, Chase discusses two options for "enabl[ing] the receiving NIC to retain or recover its ability to locate RDMA headers in the presence of sequence holes, i.e., when packets arrive out of order." Chase states that "[o]ne option is for the NIC to buffer out-of-order data ..." but this approach has all of the problems that go with buffering and clearly does not suggest splitting RDMA units and including direct memory location information with the split parts that are then put in the bodies of transport segments.

Chase mentions a more relevant second option, which is "to integrate framing support into the transport, allowing the receiver to locate RDMA headers even when packets arrive out of order. Note that every packet must contain an RDMA header for this approach to be fully general." Chase discusses two ways to accomplish this; include an RDMA header in the TCP header (TCP "options" approach), and renegotiate the MTU.

First, Chase suggests "introducing a new TCP option [TCPRDMA]". The cited "[TCPRDMA]" reference corresponds to the *Cheriton* patent reference cited in the previous Office Action and discussed by Applicant in the previous Amendment. The teachings of "[TCPRDMA]" and the Cheriton patent are the same; using the *TCP header* to carry memory address information. The present claims, as before, include remote direct memory location information in the *bodies* of transport segments. Therefore, only Chase's second approach need by considered.

Chase's second suggested approach is to "constrain[] the TCP sender's selection of [transport] segment boundaries to correspond with framing boundaries [VITCP]". A review of the "[VITCP]" reference ("VI/TCP (Internet VI)", copy included) cited by Chase shows that this second approach suggested by Chase is nothing more than "requir[ing] that incoming connection establishment attempts be rejected unless the calling and called VI attributes match (e.g., Maximum Transfer Unit Size). The protocol permits downward negotiation of MTU sizes". In other words, a prerequisite of Chase's second approach is simply that PDUs be small enough to fit in a transmission unit - a TCP packet.

Nowhere does Chase discuss or suggest dividing or splitting an RDMA unit into smaller RDMA units, and providing each smaller piece with its own header that allows direct memory placement.

In sum, the Feldmeier-Chase combination breaks RDMA packets into chunks that can be reassembled ("processed") out of order. The smaller RDMA chunks of Feldmeier-Chase cannot individually be placed directly in memory because they do not include any remote memory location information.

Withdrawal of the rejection is respectfully requested.

Improper Combination

The rejection proposes combining Chase with Feldmeier. The motive given for the combination is that adding Chase to Feldmeier "would enable placing received data in a correct memory buffer directly thereby avoiding problems such as copy buffer size requirements (Chase, section 1, Introduction)". But Chase is capable of solving this problem without Feldmeier. The rejection does not explain why it would be desirable to add Chase to Feldmeier. The rejection does not explain what problem Chase would solve for Feldmeier, or how Feldmeier would be improved by the addition of Chase. The motive provided for the combination is really only a restatement of the problem solved by Chase, namely, the problem of overhead involved with buffer to buffer copies. But Chase solves this problem on its own (without Feldmeier) by introducing RDMA, where direct copy information is added to the transport stream.

In sum, the rejection does not explain why it would have been desirable or beneficial to add Chase to Feldmeier. The rejection explains that Chase solves a problem, but the rejection does not explain the relation of Chase to Feldmeier. Applicant again notes, as discussed above, that Feldmeier is concerned with reassembling fragments (chunks) independent of the order in which they are received. That is, Feldmeier is concerned with reconstructing an upper layer packet or segment, not placing data directly into memory. Withdrawal of the rejection is respectfully requested.

DEPENDENT CLAIMS

The dependent claims are deemed to be patentable based on their dependence from allowable independent claims. The dependent claims are also independently

patentable. For example, claim 2 recites "limiting an upper layer protocol data unit size to the smaller of a maximum transport segment size and a size that will fit within non self-describing segment ". The prior art combination does not discuss or suggest this feature. Withdrawal of the rejection of the dependent claims is respectfully requested.

CONCLUSION

Accordingly, in view of the above remarks it is submitted that the claims are patentably distinct over the prior art and that all the rejections to the claims have been overcome. Reconsideration and reexamination of the above Application is requested. Based on the foregoing, Applicant respectfully requests that the pending claims be allowed, and that a timely Notice of Allowance be issued in this case. If the Examiner believes, after this Amendment, that the application is not in condition for allowance, the Examiner is requested to call the Applicant's representative at the telephone number listed below.

Application Number: 10/016,609 Attorney Docket Number: 164147.01

If this Amendment is not considered timely filed and if a request for an extension of time is otherwise absent, Applicant hereby requests any necessary extension of time. If there is a fee occasioned by this Amendment, including an extension fee that is not covered by an enclosed check please charge any deficiency to Deposit Account No. 50-0463.

Respectfully submitted,

Microsoft Corporation

Date: 18 OCT 2005

By: James D Strom

James T. Strom, 48,702
Attorney for Applicants
Direct telephone (425) 706-0362
Microsoft Corporation
One Microsoft Way
Redmond WA 98052-6399

CERTIFICATE OF MAILING OR TRANSMISSION [37 CFR 1.8(a)]

I hereby certify that this correspondence is being:

☐ deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: Mail Stop_____, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450

☒ transmitted by facsimile on the date shown below to the United States Patent and Trademark Office at (571) 273-8300.

18 OCT 2005
Date

James D Strom
Signature

James T. Strom
Type or Print Name